In the Claims:

Claim 1 (currently amended): A processor comprising:

a first plurality of threads, each of said first plurality of threads comprising one of a second plurality of processing units;

a fourth plurality of instruction packets, wherein each of said fourth plurality of instruction packets comprises a third plurality of issue groups;

each of said first plurality of threads receiving a respective one of said third plurality of issue groups from a respective one of said fourth plurality of instruction packets; and

a respective one of said second plurality of processing units executing said third plurality of issue groups in a single clock cycle, wherein at least two issue groups of said third plurality of issue groups are from different instruction packets of said fourth plurality of instruction packets.

Claim 2 (original): The processor of claim 1 wherein each of said first, second, third, and fourth pluralities is equal to two.

Claim 3 (previously presented): The processor of claim 1 wherein each of said fourth plurality of instruction packets comprises two issue groups.

Claim 4 (original): The processor of claim 3 wherein each of said fourth plurality of instruction packets is 128 bits wide.

Claim 5 (original): The processor of claim 4 wherein a first one of said two issue groups is 64 bits wide and a second one of said two issue groups is 48 bits wide.

Claim 6 (original): The processor of claim 4 wherein a first one of said two issue groups is 48 bits wide and a second one of said two issue groups is 64 bits wide.

Claim 7 (original): The processor of claim 1 wherein each of said fourth plurality of instruction packets resides in a respective instruction cache and is addressed by a respective program counter.

Claim 8 (original): The processor of claim 1 wherein each of said first, second, third, and fourth pluralities is equal to four.

Claim 9 (currently amended): A method for improving performance of a VLIW processor comprising:

dividing a first instruction packet into a first packet first issue group and a first packet second issue group;

dividing a second instruction packet into a second packet first issue group and a second packet second issue group;

providing said first packet first issue group to a first thread having a first thread processing unit and said second packet first issue group to a second thread having a second thread processing unit during a first clock cycle; and

providing said first packet second issue group to said first thread having said first thread processing unit and said second packet second issue group to said second thread having said second thread processing unit during a second clock cycle, wherein said first instruction packet is a different instruction packet than said second instruction packet.

Claim 10 (canceled).

Claim 11 (original): The method of claim 9 wherein each of said first and second instruction packets consists of 128 bits.

Claim 12 (previously presented): The method of claim 11 wherein said first packet first issue group comprises 64 bits and said first packet second issue group comprises 48 bits.

Claim 13 (previously presented): The method of claim 11 wherein said first packet first issue group comprises 48 bits and said first packet second issue group comprises 64 bits.

Claim 14 (previously presented): The method of claim 11 wherein said second packet first issue group comprises 64 bits and said second packet second issue group comprises 48 bits.

Claim 15 (previously presented): The method of claim 11 wherein said second packet first issue group comprises 48 bits and said second packet second issue group comprises 64 bits.

Claim 16 (original): The method of claim 9 wherein each of said first and second instruction packets consists of 256 bits.

Claim 17 (previously presented): The method of claim 16 wherein said first packet first issue group comprises 128 bits and said first packet second issue group comprises 112 bits.

Claim 18 (previously presented): The method of claim 16 wherein said first packet first issue group comprises 112 bits and said first packet second issue group comprises 128 bits.

Claim 19 (previously presented): The method of claim 16 wherein said second packet first issue group comprises 128 bits and said second packet second issue group comprises 112 bits.

Claim 20 (previously presented): The method of claim 16 wherein said second packet first issue group comprises 112 bits and said second packet second issue group comprises 128 bits.

Claim 21 (currently amended): A method for improving performance of a VLIW processor comprising:

dividing each one of a first plurality of instruction packets into a second plurality of issue groups;

providing each one of said second plurality of issue groups, in one of a third plurality of clock cycles, to a respective thread having a respective processing unit:

executing said first plurality of instruction packets in said third plurality of clock cycles, wherein an issue group from each one of said first plurality of instruction packets is executed in one of said third plurality of clock cycles, wherein at least two issue groups

of said second plurality of issue groups are from different instruction packets of said first plurality of instruction packets.

Claim 22 (original): The method of claim 21 wherein each of said first, second, and third pluralities is equal to two.

Claim 23 (previously presented): The method of claim 21 wherein each one of said first plurality of instruction packets comprises two issue groups.

Claim 24 (original): The method of claim 23 wherein each one of said first plurality of instruction packets is 128 bits wide.

Claim 25 (original): The method of claim 24 wherein a first one of said two issue groups is 64 bits wide and a second one of said two issue groups is 48 bits wide.

Claim 26 (original): The method of claim 24 wherein a first one of said two issue groups is 48 bits wide and a second one of said two issue groups is 64 bits wide.

Claim 27 (original): The method of claim 21 wherein each one of said first plurality of instruction packets resides in a respective instruction cache and is addressed by a respective program counter.

Claim 28 (original): The method of claim 21 wherein said first plurality is equal to four, and wherein each of said second and third pluralities is equal to two.